

DEC 02 2008

In the Claims:

Please amend claims 1, 2, 4-8, and 10, cancel claim 9, and add new claims 11-21, all as shown below.

1. (Currently Amended): A computer program product for execution by a server computer for implementing a two-phase commit protocol, comprising:

computer code for dispatching a first ~~two-phase commit protocol~~ prepare operation from a first server thread to a second server thread, wherein the first two-phase commit protocol prepare operation is associated with a first local resource participant and a first prepare phase of two-phase commit protocol;

computer code for processing a second ~~two-phase commit protocol~~ prepare operation by the first server thread in parallel to the first prepare operation being processed by the second server thread, wherein the second two-phase commit protocol prepare operation is associated with a second local resource participant and the first prepare phase of two-phase commit protocol; and

computer code for determining that the prepare phase the first two-phase commit protocol operation is complete;

computer code for dispatching a first commit operation from the first server thread to a third server thread, wherein the first commit operation is associated with the first local resource participant and a commit phase;

computer code for processing a second commit operation by the first server thread in parallel to the first commit operation being processed by the third server thread, wherein the second commit operation is associated with the second local resource participant and the commit phase; and

computer code for writing results of the commit phase to a transaction log after determining that the commit phase is complete.

2. (Currently Amended): The computer program product of claim 1 further comprising:
computer code for selecting an idle server thread to process the first ~~prepare two-phase commit protocol~~ operation.
3. (Original): The computer program product of claim 2, wherein computer code for selecting includes:
computer code for determining available server threads in a server.
4. (Currently Amended): The computer program product of claim 3 wherein a thread pool manager determines the available server threads in the server.

5. (Currently Amended): The computer program product of claim 1 further comprising:
computer code for reporting results of the prepare phase first and second two-phase commit protocol to a log.

6. (Currently Amended): A computer program product for execution by a server computer for processing two or more two-phase commit protocol operations, comprising:

computer code for processing a plurality of two or more two-phase commit protocol prepare operations in a first server thread, wherein each of the plurality of prepare operations are associated with a prepare phase and a local resource participant, wherein the computer code for processing for each prepare operation of one or more of the prepare operations two-phase commit protocol operation of the two or more two-phase commit protocol operations, except for a remaining two-phase commit protocol operation, includes:

computer code for dispatching the prepare two-phase commit protocol operation to a second server thread if a second server thread is determined to be available; and

computer code for processing the prepare two-phase commit protocol operation in the first server primary thread if no second server thread is determined to be available; and

computer code for processing the a remaining prepare two-phase commit protocol operation in the first server thread;

computer code for determining that the prepare phase is complete;

computer code for processing a plurality of commit operations, wherein each of the plurality of commit operations are associated with a commit phase and a local resource participant, wherein the computer code for processing for each commit operation of one or more of the commit operations includes:

computer code for dispatching the commit operation to a second server thread if a second server thread is determined to be available;

computer code for processing the commit operation in the first server thread if no second server thread is determined to be available;

computer code for processing a remaining commit operation in the first server thread; and

computer code for writing results of the commit phase to a transaction log after determining that the commit phase is complete.

7. (Currently Amended): The computer program product of claim 6 wherein computer code for dispatching the each two-phase commit protocol operation to a second server thread includes:

computer code for determining available server threads in a server; and

computer code for selecting one of the available server threads to be the second server thread.

8. (Currently Amended): The computer program product of claim 7 wherein a thread pool manager determines the available server threads in the server.
9. (Cancelled)
10. (Currently Amended): The computer program product of claim 6 further comprising:
computer code for reporting results of the plurality of prepare operations associated with the prepare phase ~~first two phase commit protocol operation and the second two phase commit protocol operation~~ to a log.
11. (New): The computer program product of claim 1, wherein a dedicated thread pool is used for parallel transaction operations.
12. (New): The computer program product of claim 1, wherein a transaction manager implements Java Transaction API.
13. (New): The computer program product of claim 1, wherein the first resource is an XA resource.
14. (New): The computer program product of claim 6, wherein each prepare operation of one or more of the prepare operations is associated with an XA resource.
15. (New): The computer program product of claim 6, wherein a dedicated thread pool is used for parallel transaction operations.
16. (New): The computer program product of claim 6, wherein a transaction manager implements Java Transaction API.
17. (New): The computer program product of claim 1, wherein all of the prepare operations and all of the commit operations are part of a single transaction.
18. (New): The computer program product of claim 6, wherein all of the prepare operations and all of the commit operations are part of a single transaction.

19. (New): A method, comprising:

processing a plurality of prepare operations, wherein each of the plurality of prepare operations are associated with a prepare phase and a local resource participant, wherein the computer code for processing for each prepare operation of one or more of the prepare operations includes:

dispatching the prepare operation to a second server thread if a second server thread is determined to be available; and

processing the prepare operation in the first server thread if no second server thread is determined to be available;

processing a remaining prepare operation in the first server thread;

determining that the prepare phase is complete;

processing a plurality of commit operations, wherein each of the plurality of commit operations are associated with a commit phase and a local resource participant, wherein the computer code for processing for each commit operation of one or more of the commit operations includes:

dispatching the commit operation to a second server thread if a second server thread is determined to be available;

processing the commit operation in the first server thread if no second server thread is determined to be available;

processing a remaining commit operation in the first server thread; and

writing results of the commit phase to a transaction log after determining that the commit phase is complete.

20. (New): The method of claim 19, wherein a transaction manager implements Java Transaction API.

21. (New): The method of claim 19, wherein the first resource is an XA resource.